公共卫生事件中医疗数据访问控制与安全共享研究

韩 刚[1,2], 王嘉乾[1,2], 罗 维[1], 吕英泽[1]

[1]西安邮电大学 网络空间安全学院 西安 中国 710121
[2]无线网络安全技术国家工程实验室 西安 中国 710000

摘要 随着新冠疫情的持续发展, 许多国家和地区都对确诊患者及密接者的个人信息数据和位置数据进行了严密的监管。与此同时, 如何在共享患者必要信息的同时, 确保患者及密接者的个人隐私不被泄露, 访问过程透明化、可溯源、数据不被篡改, 已成为当今亟需解决的关键问题。基于此, 本文提出了一种可追责的医疗属性通行证(AMAP)访问控制方案, 方案首先将区块链与基于属性的访问控制模型相结合, 在引入区块链对访问过程进行溯源的同时, 将访问控制策略和访问时系统中的关键步骤以智能合约的形式部署到区块链上, 使整个系统既能保障用户对数据的安全访问, 又能够对整个访问过程进行溯源。特别地, 方案引入了医疗属性通行证模块, 用户以通行证的方式申请访问, 避免了传统访问控制模型中主体属性与访问控制策略的多次匹配, 在实现医疗数据细粒度访问控制的同时, 一定程度上提高了访问效率。最后, 通过安全性分析表明本方案可以抵抗拒绝服务攻击、恶意篡改攻击、单点失效攻击、主体伪装攻击、重放攻击等。实验及性能分析表明本方案与其他方案相比, 在相同访问控制策略的情况下访问次数越多, 本方案的优势越明显; 在相同访问次数情况下访问控制策略个数越多, 本方案的优势越明显。

关键词 信息安全; 区块链; AMAP; 细粒度; 访问控制
中图法分类号 TP309.2 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2023.01.04

Research on Medical Data Access Control and Security Sharing in Public Health Events

HAN Gang[1,2], WANG Jiaqian[1,2], LUO wei[1], LV Yingze[1]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2] National Engineering Laboratory for Wireless Network Security Technology, Xi'an 710000, China

Abstract With the continuous developing of the COVID-2019, more and more countries and regions have strictly supervised the personal information and location data of confirmed patients and their close contacts. At the same time, how to share the necessary information of patients while ensuring that the personal privacy of patients and their close contacts is not leaked, the access process is transparent, traceable, and data is not tampered with, has become a key issue that needs to be solved urgently. Based on this, we propose an accountable medical attribute pass (AMAP) access control scheme in this paper. The scheme first combines the blockchain with an attribute-based access control model. While introducing the blockchain to trace the source of the access process, the access control strategy and key steps in the access system are deployed on the blockchain in the form of smart contracts, so that the entire system can not only ensure the safe access of users to data, but also trace the source of the entire access process. In particular, the solution introduces the medical attribute pass module. Users apply for access in the form of a pass, which avoids multiple matches between subject attributes and access control strategies in the traditional access control model. While achieving fine-grained access control to medical data, a certain degree Improved access efficiency. Finally, the security analysis shows that this scheme can resist denial of service attacks, malicious tampering attacks, single point of failure attacks, main body masquerading attacks, replay attacks, etc. Experiments and performance analysis show that this solution is compared with other solutions. Under the same access control strategy, the more access times, the more obvious the advantages of this solution; the more access control strategies have the same access control strategy, the more effective the solution is. The more obvious the advantages.

Key words information security; blockchain; AMAP; fine-grained; access control

通讯作者: 王嘉乾, 本科, Email: wangjq990817@163.com。
本课题得到国家自然科学基金(No. 62102312), 陕西省自然科学基础研究计划资助项目(No. 2021JQ-722), 陕西省高校科协青年人才托举计划(No. 20210119), 陕西省教育厅科研计划项目(No. 20JK0906)资助。
收稿日期: 2021-09-27; 修改日期: 2021-12-20; 定稿日期: 2022-11-03

1 引言

2020 年, 全球爆发了新型冠状病毒(COVID-2019), 疫情瞬间肆虐全球。截至到 2021 年 9 月 8 日, 全球已经有 2.23 亿确诊病例, COVID-2019 已经成为世界上已知的最严重的突发公共卫生事件。由于 COVID-2019 具有传染性强, 变体病毒种类多的特点, 如果对出现的患者不进行即使管控, 将对国家和社会的正常运转带来重大影响。因此, 相关医疗机构及时掌握患者及其密切接触者的个人信息数据和位置数据便成为防疫过程中最为重要的一环。而在掌握相关患者及其密切接触者的个人信息数据和位置数据的过程中, 如何保证相关人员在共享必要位置信息的同时, 确保患者以及密切接触者的个人隐私不被泄露, 便成为当今急需研究和解决的一个关键问题。

传统医疗信息系统中, 数据信息由不可信第三方进行保管, 不可信第三方可以对数据进行任意的伪造、篡改和泄露, 同时没有对信息系统的访问控制权限进行精细的划分, 因此容易被医疗系统之外的成员恶意查看和泄露关键医疗信息。另外, 整个医疗信息系统访问过程不透明、不可溯源, 如果遇到关键医疗信息被攻击者恶意窃取、损坏和泄露的问题[1-2], 将无法对破坏者进行有效追责。

以上这些传统医疗信息系统中存在的问题给医疗信息的共享和患者个人隐私的保护埋下了隐患。

医疗信息系统中的信息, 例如患者病历、医生信息、患者个人信息, 是涉及相关人员隐私和安全的重要数据, 而数据访问控制[3]技术是保护此类数据的关键技术。当前已有研究者针对该技术展开了大量研究, 张怡婷等[4]提出了一种结合基于策略的权限控制(PBAC)模型和 IBE 加密技术的访问控制方案, 利用患者 ID, 条件访问位作为 IBE 公钥对数据进行加密, 只有满足条件的数据访问者才可以获得加密密钥和加密数据, 实现了细粒度的访问控制。Deng 等[5]设计了一种基于角色的访问控制(RBAC)[6-7]的医疗云平台, 达到了对医疗云平台后台访问控制的目的; 苗等[8]在基于属性加密技术实现细粒度访问控制的同时, 还引入了不经意传输技术, 利用不经意传输技术对用户属性进行匿名, 使得攻击者无法根据用户的属性来推断出其密文。以上研究在重要信息的访问控制上均取得了一定的成效, 但仍然没有解决访问过程不透明、易篡改和不可溯源的问题。因此, 仍然需要引入其他技术来解决上述问题。

随着以区块链技术[9]为依托的比特币[10]的产生与兴起, 区块链技术开始进入了人们的视野。区块链作为一种由多方共同参与的分布式数据库, 其拥有多方维护、去中心化、不可篡改、交易公开透明等优点。区块链技术被广泛应用到货币交易、物联网、人工智能、大数据[11]等领域。而在与访问控制技术相关的方面, 已有研究者做出了相关的研究与探索。Maesa DDF 等人[12]将区块链与访问控制策略相结合, 但区块链只是作为访问控制策略管理的数据库, 同时还需要第三方提供访问控制服务; Thwin 等[13]提出一种基于区块链的访问控制模型, 该模型中使用了云存储和私有链, 同时还包括代理重加密等加密机制; YANG 等[14]将区块链技术与属性加密相结合, 使得其在多对多通信中共享医疗数据的同时, 能够实现细粒度访问控制以及验证医疗数据源的真实性; 葛纪红等[15]将区块链技术与访问控制技术相结合, 在进行细粒度访问控制的同时, 将数据的数据摘要以 hash 值的形式存入了区块链中, 以保证数据的不可篡改。但却无法保证整个访问过程的公开可溯源; Zyskind 等[16]将区块链链上与链下数据存储相结合, 实现了个人的数据管理, 但却只能实现数据的读操作。Neisse 等[17]为了提高数据使用的透明度和访问效率, 在区块链上部署了公开审计合同, 同时能够实现记账和溯源功能。刘敖迪等[18]通过对传统的基于属性的访问控制模型(ABAC)[19-20]进行改进, 引入了 BBAC-BD 模型, 将访问控制技术与区块链技术相结合, 摆脱了传统访问控制集中管理可能出现的由于单点故障导致系统崩溃的问题和访问控制透明度的问题, 并将其引入大数据领域。但没有给出具体的访问控制流程, 并且仍然没有解决访问过程可溯源的问题; 谢绒娜等[21]提出了一种基于区块链的可溯源的访问控制机制, 引入了基于属性的访问控制模型并通过加入区块链技术对其进行了改进, 使得其访问控制得以溯源, 并且访问过程公开透明。但该方案在访问时需要在每次访问时都要对用户的属性和访问控制策略进行匹配, 而这无疑增加了系统负担。

针对这些问题, 本文通过对传统基于属性的访问控制模型进行改进, 在传统基于属性访问控制的基础上, 引入属性访问通行证, 使主体在限定时间内进行访问时, 无需再次对属性和访问控制策略进行匹配, 使其能够在保证访问控制完整性和严密性的前提下, 减小系统开销。同时, 本文将访问控制技术与区块链技术相结合, 以解决数据不可篡改、访问公开透明、访问过程不可溯源的问题。

另外, 由于医疗系统中需要采用的区块链为联盟链[22], 因此本方案不再使用传统的 POX[23]系列共

识算法, 而是使用实用拜占庭容错算法, 以实现通过较少节点便可以达成共识的目的。

2 基础知识

2.1 RSA 算法

RSA 属于一种公钥密码算法, 是 1977 年由罗纳德·李维斯特(Ron Rivest)、阿迪·萨莫尔(Adi Shamir)和伦纳德·阿德曼(Leonard Adleman)一起提出的。1987 年首次公布, RSA 就是他们三人姓氏开头字母拼在一起组成的。RSA 是目前最有影响力的公钥加密算法, 它能够抵抗到目前为止已知的绝大多数密码攻击, 已被 ISO 推荐为公钥数据加密标准。

RSA 算法的具体描述如下:

(1) 任意选取两个不同的大素数 p 和 q 计算乘积

$$n = pq, \varphi(n) = (p-1)(q-1) \tag{1}$$

(2) 任意选取一个大整数 e, 满足:

$$\gcd(e, \varphi(n)) = 1 \tag{2}$$

整数 e 用做加密钥;

(3) 确定的解密钥 d, 满足:

$$(de) \bmod \varphi(n) = 1 \tag{3}$$

即:

$$de = k\varphi(n) + 1, k \geqslant 1 \tag{4}$$

是一个任意的整数;

(4) 公开整数 n 和 e, 秘密保存 d;

(5) 将明文 m ($m < n$ 是一个整数)加密成密文 c, 加密算法为:

$$c = E(m) = m^e \bmod n \tag{5}$$

(6) 将密文 c 解密为明文 m, 解密算法为:

$$m = D(c) = c^d \bmod n \tag{6}$$

2.2 基于属性的访问控制(ABAC)模型

步骤 1: 数据拥有者将数据即客体上传到数据服务器中。

步骤 2: 数据拥有者针对对应的客体生成访问控制策略, 并将访问控制策略上传到策略管理点, 然后策略管理点将访问控制策略上传到策略控制库中。

步骤 3: 数据使用者即主体向策略执行点发出访问请求, 然后策略执行点生成访问决策请求, 并把访问决策请求上传到策略决策点。

步骤 4: 策略决策点在收到访问决策请求后, 首先向策略管理点发出策略查询请求, 然后策略管理点在策略控制库中查询访问控制策略, 并将查询到的访问控制策略返回给策略决策点。

步骤 5: 策略决策点向策略信息点发出属性请求查询, 然后策略信息点将相应的属性返回给策略决策点。

步骤 6: 策略决策点将收到的访问控制策略和属性进行匹配和整合, 然后生成访问控制结果, 并将访问控制结果返回给策略执行点。

步骤 7: 策略执行点将收到的访问控制结果上传至数据库服务器, 数据服务器对收到的客体请求即访问控制结果与相应的客体进行匹配, 然后将客体发送给策略执行点。

步骤 8: 策略执行点将客体发送给主体。

流程图如下:

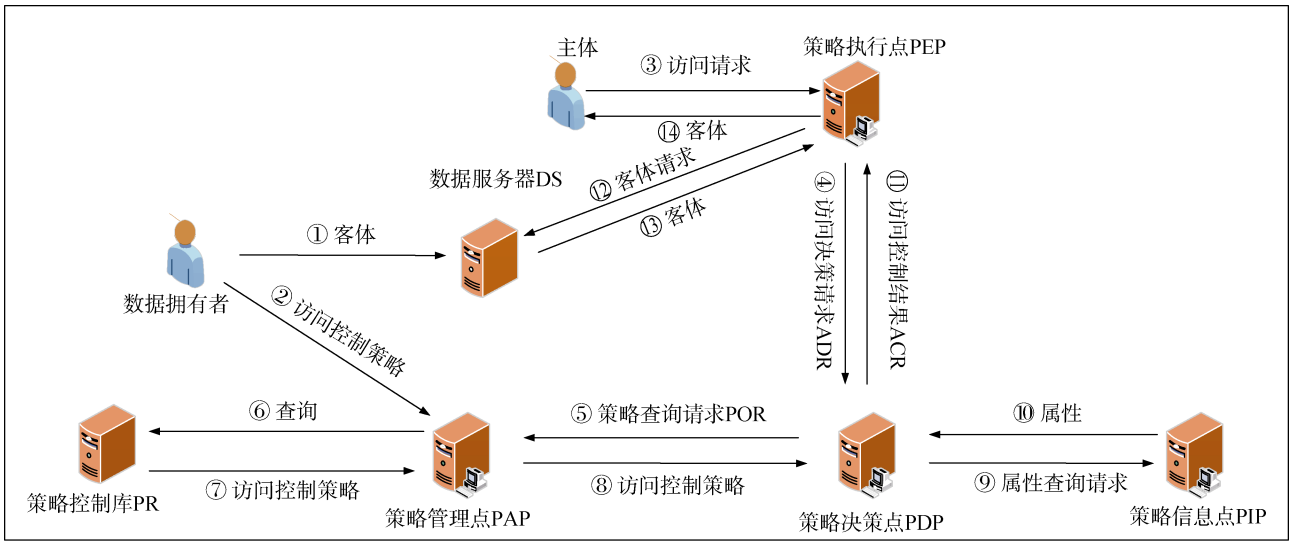

图 1 基于属性的访问控制模型流程

Figure 1 Attribute-based access control model process

2.3 区块链技术

区块链技术首次被提出于中本聪的论文《比特币: 一种点对点的电子现金系统》[9], 其最初的设计目标是为了设计一种不需要不可信第三方参与便可以进行交易的数字货币, 而其本身是一种类似于链表结构的数据结构。随着比特币的出现, 区块链作为一种新兴的技术, 凭借着其去中心化, 可溯源, 不可篡改等优点, 逐渐被应用于大数据, 物联网, 智慧医疗, 智慧城市[23-26]等各种场景。

区块链的基础框架主要包括: (1)基础网络层下的数据层和网络层, 用来存储交易数据, 加盖时间戳以及对交易信息进行广播等; (2)中间协议层下的共识层, 激励层和合约层。其中共识层主要是对交易达成共识, 形成新的区块并将新的区块加入到区块链中。激励层是记录记账节点的"薪酬体系", 能够对记账节点产生一定的激励。而合约层主要是用来搭载智能合约, 其中记录和规定交易过程中的各种规范和条件, 一旦达到某一条件, 合约层中的智能合约便会开始运行。区块链框架结构如下图:

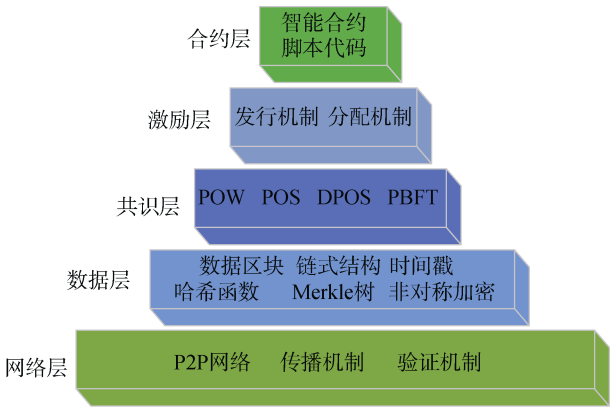

图 2 区块链框架结构

Figure 2 Blockchain framework structure

2.4 实用拜占庭容错共识机制(PBFT)

实用拜占庭容错算法是联盟链中一种共识算法, 由麻省理工学院的 Miguel Castro 和 Barbara Liskov 于 1999 年提出, 解决了原始拜占庭容错算法效率不高的问题, 算法复杂度不再是 a^x 级别, 而减小成了多项式级别, 使得拜占庭容错算法在实际系统应用中变得可行。

利用实用拜占庭容错算法可以解决在有限个节点情况下的拜占庭将军问题, 当联盟链中的恶意节点为 f 时, 只要链中的总节点个数 $\geq 3f+1$, 便可以保证共识的达成。

在实用拜占庭容错共识机制中, 运行过程一共分为 5 个阶段:

请求(request): 用户将请求发送给主节点。

预准备(pre-prepare): 主节点收到消息后向, 生成预准备消息, 并将消息广播到全体副节点。

准备(prepare): 次节点在收到主节点发送的预准备消息后, 生成准备消息, 并将准备消息广播到全网节点。

确认(commit): 副节点首先收到其他副节点的准备消息, 并与主节点发送的预准备消息对比进行确认。若副节点收到除自己广播外的 $2f$ 个消息, 并全部验证通过时, 副节点便生成确认消息, 并向全网广播。

响应(reply): 当用户从网络中收到 $f+1$ 个相同的消息时, 说明共识达成, 数据将被写入区块中, 并将新的区块链接到整个区块链上。共识过程如下:

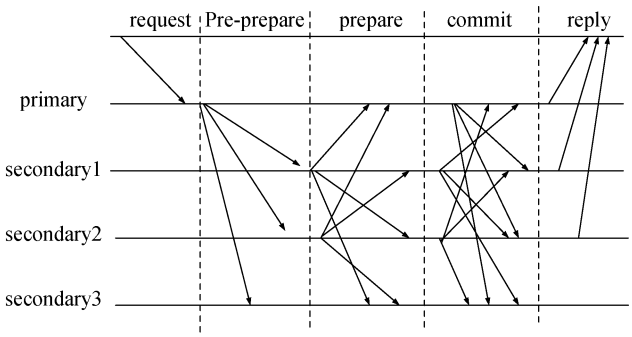

图 3 PBFT 共识过程

Figure 3 PBFT consensus process

2.5 智能合约

"智能合约"的概念可以追溯到 1995 年, 由尼克·萨博(Nick Szabo)提出。智能合约是一种规则公开透明, 不可篡改, 合约内的规则以及数据对外部可见电子合约。它被定义为一种以代码形式规定的协议, 该协议通过计算机编写的程序或其他电子设备的形式实现。伴随着区块链的发展, 现阶段已经实现了可编程合约的技术, 让智能合约从理论变成了现实, 当满足合约所需要的条件时, 将自动启动智能合约的代码, 同时区块链自带的共识算法能够构建出一套状态机系统, 可以使智能合约能够高效的运行。目前, 智能合约在金融领域, 医疗领域, 自动驾驶领域等多个重要领域均有着广阔的前景。

3 AMAP 方案

针对上述传统访问控制模型在医疗系统中面临的挑战, 本文提出了一种可追责的医疗属性通行证访问控制(Accountable medical attribute pass access

control, AMAP)方案, 将不可信第三方以智能合约的形式部署在区块链上, 以实现访问控制过程的去中心化。另外, 将数据以数据摘要的形式存储在区块链上, 避免了区块链存储空间较小的问题, 同时保证了数据的不可篡改性, 引入日志记录点对整个访问控制过程进行记录, 使其可溯源、可追责。

3.1 系统架构

(1) 通行证属性管理中心(Token Attribute Management Center, TAMC): 主体 S(本文中主体指公共卫生事件患者和相关医师等需要访问相关医疗数据的人)进行注册申请时, 通行证管理中心 TAMC 对主体 S 的身份信息进行判断, 然后根据主体 S 的身份信息, 为主体 S 分发相应的属性通行证 T。在主体 S 的通行证失去时效性后对通行证进行撤销, 同时将授权和撤销的过程生成日志, 将日志记录在日志区块链中, 保证通行证分发的透明和可溯源, 可以避免 TAMC 被恶意攻击或控制造成身份与属性不匹配, 从而防止数据被恶意访问。通行证属性管理中心 TAMC 还可以对属性发布方 AP 上传的属性进行整合, 并将其发送至日志部署点。除此之外, 通行证属性管理中心 TAMC 能够接收到密钥分发中心 KDC 分发到的主体公钥, 并利用主体公钥对主体所申请的通行证进行加密。

(2) 属性发布方(Attribute Publisher, AP): 对医疗系统中所需属性进行判定和设置, 并将设置好的属性上传至通行证属性管理中心。

(3) 密钥分发中心(Key Distribution Center, KDC): 为申请的主体生成公私钥对, 并分别将公钥和私钥分发给通行证属性管理中心 TAMC 和主体 S。

(4) 日志部署点(Log Deployment Point, LDP): 将整个客体上传与访问授权过程记录成日志, 并将日志部署到日志链上。

(5) 数据部署点(Data Deployment Point, DDP): 将数据拥有者上传的客体摘要以及存储地址部署到客体链上。

(6) 策略部署点(Policy Layout Point, PLP): 将策略管理点整合好的访问控制策略部署到客体链上。

(7) 策略管理点(Policy Administration Point, PAP): 将数据拥有者上传的访问控制策略进行整合并将其发送到策略部署点。同时接收策略决策点上传的属性信息并将属性信息与访问控制策略进行匹配, 并将匹配结果发送给策略决策点。

(8) 策略执行点(Policy Enforcement Point, PEP): 对主体上传的通行证进行验证, 并记录通行证的 *token_id* 并将验证成功的通行证发送到策略决策点。同时接收由访问授权点发送的访问凭据, 并将访问凭据发送给主体和日志部署点。

(9) 策略决策点(Policy Decision Point, PDP): 接受策略执行点上传的属性通行证, 提取出通行证中的 *token_type*, 然后将 *token_type* 发送给策略管理点。在接收到策略管理点的访问控制策略之后, 再将访问控制策略发送给访问授权点。

(10) 访问授权点(Access Authorization Point, AAP)在收到访问决策之后, 生成访问凭据, 并将访问凭据发送给策略执行点。

在以上架构中, 日志部署点 LDP, 数据部署点

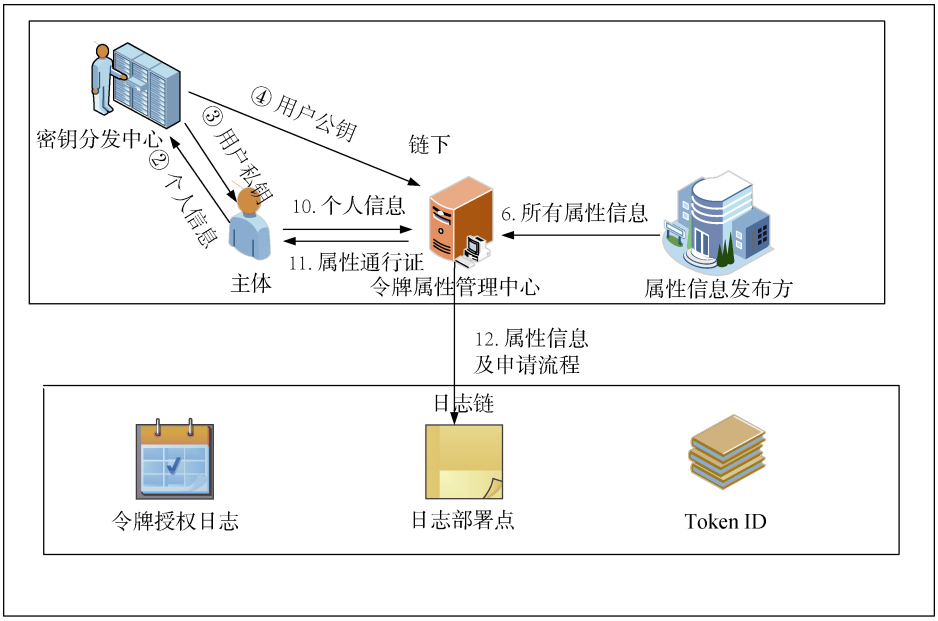

图 4 通行证申请流程

Figure 4 Token application process

DDP, 策略部署点 PLP, 策略执行点 PEP, 策略决策点 PDP, 访问授权点 AAP 均以智能合约的形式部署到区块链上, 当达到上述所需条件时, 便自动执行, 在一定程度上避免了恶意攻击与篡改。

3.2 系统流程

3.2.1 通行证申请

步骤 1: 首先由属性发布方 AP 将医疗系统中所需要的所有属性上传至通行证属性管理中心 TAMC, 通行证属性管理中心 TAMC 对所有属性进行整合并将其发送到日志部署点 LDP, 然后日志部署点将属性上传至日志链 LC(Log Chain)中, 以保证所有属性的公开透明。

步骤 2: 当主体 S 申请属性通行证时, 主体 S 将自己的个人信息上传至通行证属性管理中心 TAMC, 通行证属性管理中心 TAMC 对主体信息与属性进行匹配, 然后利用匹配好的属性生成相应的属性通行证 token。

步骤 3: 主体 S 将个人信息上传至密钥分发中心 KDC, 密钥分发中心 KDC 利用主体上传的个人信息, 对应生成该主体 S 的公私钥对, 将私钥发送给主体 S, 将公钥发送给通行证属性管理中心 TAMC。

步骤 4: 通行证属性管理中心 TAMC 生成通行证 token 后, 利用主体 S 的公钥对通行证进行加密并将通行证发送给主体 S。同时通行证属性管理中心 TAMC 将整个通行证申请过程以及生成通行证的 *token_id* 生成日志, 并将日志发送到日志部署点 LDP, 日志部署点 LDP 将日志记录到日志链 LC 中。

3.2.2 客体上传

步骤 1: 数据拥有者 DO 首先将所拥有的客体 O 上传至数据库服务器 DB 中, 数据库服务器 DB 接收到客体 O 之后, 对客体 O 进行 hash 生成客体摘要 OH(Object Hash), 然后将客体摘要 OH 与客体存储的地址 A(Address)发送给数据拥有者 DO, 数据拥有者 DO 拿到客体摘要 OH 和地址 A 后, 将数据 D 上传至数据部署点 DDP, 数据部署点 DDP 将数据 D 上传到客体链 OC(Object Chain)中。

步骤 2: 数据拥有者 DO 将制定好的访问控制策略 ACS(Access Control Strategy)上传到策略管理点 PAP, 策略管理点 PAP 对策略进行整合并将策略发送到策略部署点 PLP, 策略部署点 PLP 将访问控制策略 ACS 部署到客体链 OC 上, 然后将策略在客体链 OC 上的地址发送到策略管理点 PAP, 由策略管理点 PAP 对地址进行储存。

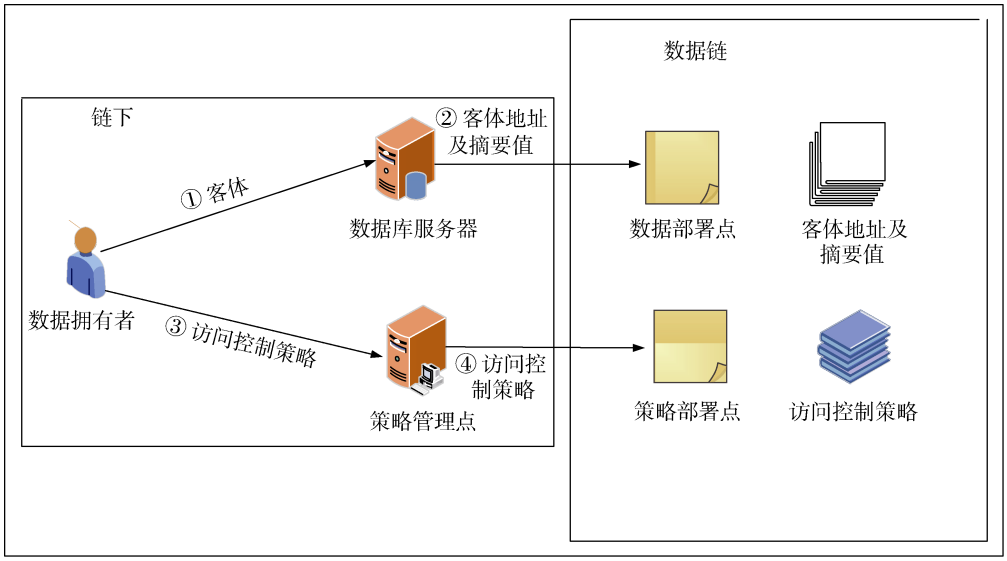

图 5 客体上传流程

Figure 5 Object upload process

3.2.3 访问授权

步骤 1: 主体 S 将自己的通行证上传至策略执行点 PEP, 由策略执行点 PEP 对通行证信息进行验证, 并记录下 *token_id*。

步骤 2: 策略执行点 PEP 将成功验证信息的通行证发送到策略决策点 PDP, 策略决策点 PDP 首先判断 *token_type*, 然后将 *token_type* 上传至策略管理点 PAP。策略管理点 PAP 根据具体的属性信息匹配相应的访问控制策略 ACS, 然后将匹配到的访问控制策略 ACS 发送给策略决策点 PDP。

步骤 3: 策略决策点 PDP 将属性信息与对应的访问控制策略 ACS 发送到访问授权点 AAP, 访问授权点 AAP 根据接收到的属性和访问控制策略 ACS 生成访问凭据 access, 并将访问凭据 access 传至策略执行

点 PEP。策略执行点 PEP 将访问凭据 access 发送给主体 S 和日志部署点 LDP。最后由日志部署点 LDP 将访问授权日志 AAL(Access Authorization Log)和访问凭据 access 部署到日志链 LC 上。

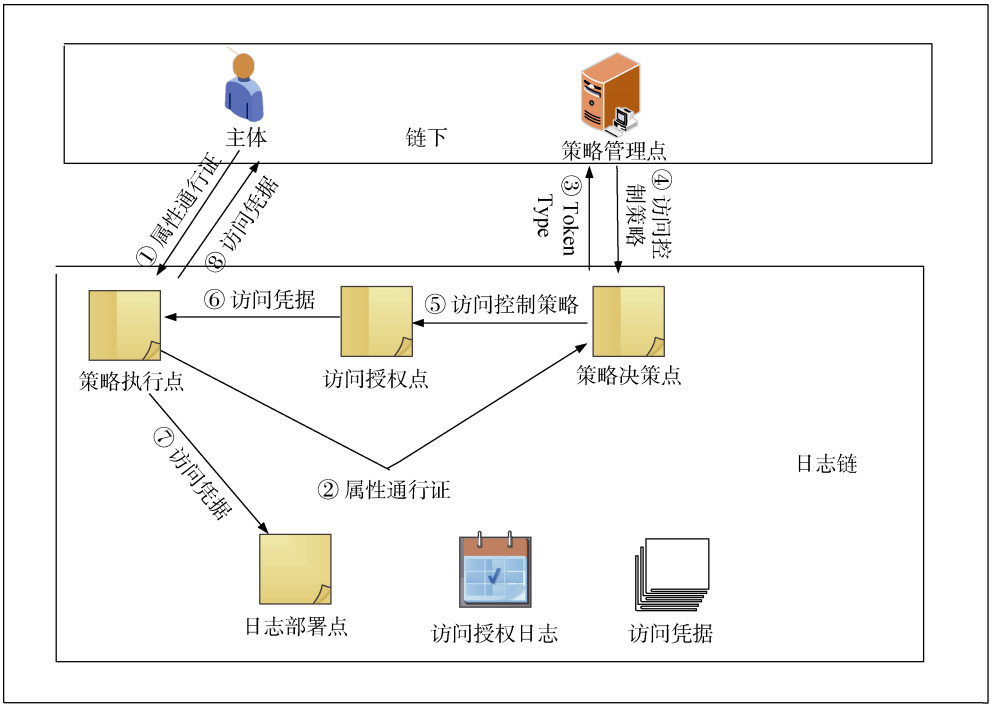

图 6　访问授权流程

Figure 6　Access authorization process

3.2.4　访问

步骤 1: 主体 S 将拿到的访问凭据 access 上传到数据库服务器 DB 中, 数据库服务器 DB 对访问凭据 access 进行验证并判断验证结果, 如果验证成功, 则将对应客体 O 发送给主体 S。反之, 拒绝其访问请求。

步骤 2: 数据库服务器 DB 将访问凭据标识符 *access_id* 进行记录, 并将其上传到日志部署点 LDP, 由日志部署点 LDP 将访问凭据标识符 *access_id* 部署到日志链 LC 中。

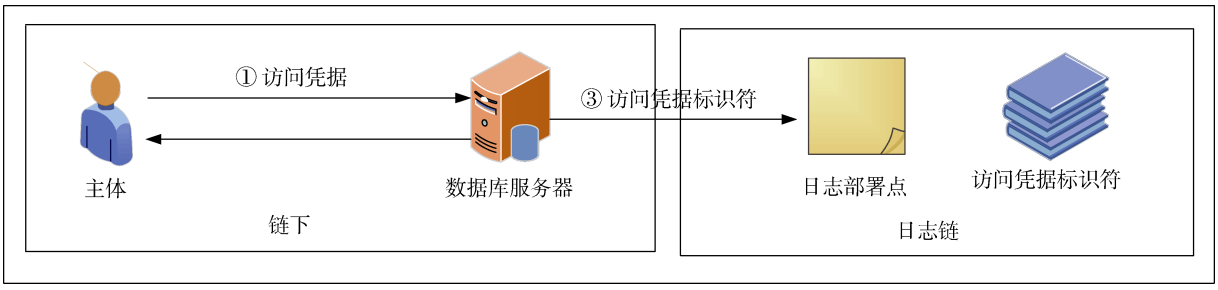

图 7　访问流程

Figure 7　Access process

3.3　智能合约的部署与实现

3.3.1　日志部署点 LDP

日志部署点 LDP 主要以智能合约的方式在通行证申请阶段, 访问授权阶段以及访问阶段发挥作用, 接受并整合需要部署到日志链上的信息, 具体过程如下:

步骤 1:

在通行证申请阶段: 当通行证属性管理中心 TAMC 将属性 Attribute_all 和主体申请流程信息 process 上传到日志部署点 LDP 的条件达成时, 日志部署点 LDP 首先得到 Attribute_all 和 process, 然后对 Attribute_all 和 process 进行整合, 最后将

Attribute_all 和 process 上传到日志链上。

Algorithm 1 : Token application stage

```
begin:
{
if TAMC.send(Attribute_all, process) == true:
{
LDP.get(Attribute_all, process)
u = LDP.arrange(Attribute_all, process)
LDP.upload(u)
}
end;
}
```

步骤 2:

在访问授权阶段: 当策略执行点 PEP 将访问凭据 access 上传到日志部署点 LDP 的条件达成时, 日志部署点 LDP 首先得到 access, 然后调对 access 进行整合, 最后将 access 上传到日志链上。

Algorithm 2: Access authorization stage(LDP)

```
begin:
{
if PEP.send(access) == true:
{
LDP.get(access)
a = LDP.arrange(access)
LDP.upload(a)
}
end;
```

步骤 3:

在访问阶段: 当数据库服务器 DB 将访问凭据的唯一确认标识符 *access_id* 上传到日志部署点 LDP 的条件达成时, 日志部署点 LDP 首先得到 *access_id*, 然后将 *access_id* 上传到日志链上。

3.3.2 数据部署点 DDP

数据部署点以智能合约的方式在客体上传阶段发挥作用, 当数据库服务器 DB 向数据部署点 DDP 发送客体摘要值 OH 和客体地址 A 的条件达成时, 数据部署点 DDP 首先得到 OH 和 A, 然后将 OH 和 A 上传到数据链上, 具体过程如下:

Algorithm 3: Visit phase

```
begin:
{
if DB.send(access_id) == true:
{
LDP.get(access_id)
LDP.upload(access_id)
}
end;
}
```

Algorithm 4: Object upload stage(DDP)

```
begin:
{
if DB.send(OH, A) == true:
{
DDP.get(OH, A)
DDP.upload(OH, A)
}
end;
}
```

3.3.3 策略部署点 PLP

策略部署点 PLP 以智能合约的方式在客体上传阶段发挥作用, 当策略管理点 PAP 向策略部署点 PLP 发送访问控制策略 ACS 条件达成时, PLP 首先得到 ACS, 然后通过将 ACS 部署到数据链上具体过程如下:

Algorithm 5: Object upload stage(PLP)

```
begin:
{
if PAP.send(ACS) == true:
{
PLP.get(ACS)
PLP.upload(ACS)
}
end;
}
```

3.3.4 策略执行点 PEP

策略执行点 PEP 以智能合约的方式在访问授权阶段发挥作用, 分别在主体 S 发送上传通行证以及访问授权点 AAP 将访问凭据 access 发送给策略执行点 PEP 的条件达成时, PEP 接收访问凭据 access 并实现对应功能, 具体过程如下:

步骤 1:

当主体 S 发送上传通行证 token 的条件达成时, PEP 首先得到通行证, 再对通行证进行验证, 若验证成功, 则对 *token_id* 进行记录, 然后将 token 发送给 PDP。

Algorithm 6: Access authorization stage(PEP)

begin:
{
if $S.send(token) == true$:
{
$PEP.get(S\_token)$
if $PEP.verify(S\_token) == true$:
{
$PEP.record(token\_id)$
$PEP.send(S\_token)$ *to PDP*
}
}
end;
}

步骤 2:

当访问授权点 AAP 发送访问凭据 access 的条件达成时, PEP 首先得到 access, 然后将 access 发送给主体 S 以及日志部署点 LDP。

3.3.5 策略决策点 PDP

策略决策点 PDP 以智能合约的形式在访问授权阶段发挥作用。分别在策略执行点 PEP 发送 S_token 的条件达成以及策略管理点 PAP 发送匹配好的访问控制策略 ACS 的条件达成时, 策略执行点 PDP 接收 S_token 和 ACS 并实现对应功能, 具体过程如下:

步骤 1:

当策略执行点PEP发送S_token的条件达成时, PDP对 *token_type* 进行判断, 并将判断结果发送给 PAP。

Algorithm 7: Access authorization stage(PDP)

begin:
{
if $PEP.send(S\_token) == true$:
{
$type = PDP.judge(token\_type)$
$PDP.send(type)$ *to PAP*
}
end;
}

步骤 2:

当策略管理点 PAP 发送匹配好的访问控制策略 ACS 的条件达成时, PDP 首先收到(*ACS,S_info*), 再将*(ACS,S_info)*发送给访问授权点 AAP。

3.3.6 访问授权点 AAP

访问授权点 AAP 以智能合约的形式在访问授权阶段发挥作用, 当 PDP 发送(*ACS,S_info*)的条件达成时, APP 生成 access 通行证, 然后将 access 通行证发送给策略执行点 PEP, 具体过程如下:

Algorithm 8: Access authorization stage(AAP)

begin:
{
if $PDP.send(ACS, S\_info) == true$:
{
$AAP.generate(access)$
$AAP.send(access)$ *to PEP*
}
end;
}

3.4 通行证数据结构

本文中通行证主要包括属性通行证(Attributes Token)和访问通行证(Access Token), 通行证最为主体申请访问和访问的唯一凭据, 是整个系统架构中的重要组成部分。属性通行证主要作用是上传主体属性并与相应的访问控制策略进行匹配并申请得到访问通行证, 而访问通行证的主要作用是申请访问数据中的相应客体。其中通行证的数据结构中的内容包括了

通行证内的关键信息。接下来本文分别对属性通行证和访问通行证的数据结构进行详细介绍。

3.4.1 属性通行证数据结构

Attributes Token 中主要有 Token Header、Token Body。Token Header 中包含了判断角色身份的关键信息, 包括 *Token ID*、*Token Type*、Asymmetric Encryption Algorithm。*Token ID* 是角色身份的唯一确认标识符, 每个 Attributes Token 都有自己唯一的 *Token ID*; *Token Type* 内主要含有有关属性的信息, 明确指明了该通行证所指向的属性类。而 Asymmetric Encryption Algorithm 中包含了加密该通行证所用到的非对称加密算法, 本文使用的是 RSA 加密算法, 具体加密流程见下文。

Token Body 中主要有 *Generate Time*、*Token Data*、*Expiration Time*。*Token Data* 中有申请人的个人信息, *Generate Time* 表示该通行证生成的时间, 而 *Expiration Time* 表示该通行证的到期时间。属性通行证数据结构表如表 1 所示:

表 1 属性通行证数据结构

Table 1 Attribute token data structure

	属性通行证字段名称	描述
通行证头	*Token ID*(通行证号)	唯一确认标识符
	Token Type(通行证类型)	通行证信息
	Encryption Algorithm (加密算法)	RSA 算法
通行证主体	*Token Data*(通行证数据)	申请通行证用户信息
	Generation Time(生成时间)	通行证生成时间
	Expiration Time(到期时间)	通行证到期时间

3.4.2 访问通行证数据结构

Access_Token 中主要有 Access Header、Access Body。Access Header 主要包括 *Access Strategy*、*Access Attributes*、*Access ID*。*Access ID* 是访问通行证的唯一确认标识符, *Access Strategy* 具体描述了具体的访问控制策略, 控制访问者所能访问的具体信息。而在 *Access Attributes* 中, 则详细解释描述了进行访问时主体所具有的属性。

在利用访问通行证访问数据过程中, 数据库服务器将对 *Access ID* 进行记录且将其上传到日志链中。而 *Access ID* 在使用过程中仅可以使用一次。若主体后续仍需对数据进行访问, 则需要再次利用属性通行证对访问凭据即访问通行证进行申请。

访问通行证数据结构如表 2 所示, Access Body 中包含的主要信息有 *Data*、*Generate Time*、*Expiration Time* 和 *Information Address*。*Data* 中含有申请该通行证的主体的信息, *Generate Time* 表示访问通行证生成的时间, *Expiration Time* 表示访问通行证到期时间。而 *Information Address* 表示访问通行证所要访问的信息在数据库服务器 DB 中的地址。

表 2 访问通行证数据结构

Table 2 Access token data structure

	访问通行证字段名称	描述
通行证头	*Access ID*(通行证号)	访问通行证唯一确认标识符
	Access Strategy(访问策略)	允许用户访问信息的范围
	Access Attribute(访问属性)	访问信息者拥有的属性
通行证主体	*Access Data*(通行证数据)	申请通行证的用户信息
	Generation Time(生成时间)	访问通行证生成时间
	Expiration Time(到期时间)	访问通行证到期时间

3.5 通行证加解密方案

3.5.1 密钥对的生成

(1): 选择一对不同的, 足够大的素数 p,q。

(2): 计算

$$n = p * q。 \tag{7}$$

(3): 计算

$$f(n) = (p-1) * (q-1)。 \tag{8}$$

(4): 找一个与 $f(n)$ 互质的数 e 作为公钥指数, 且:

$$1 < e < f(n) \tag{9}$$

(5): 计算私钥指数 d, 使 d 满足:

$$d * e \bmod f(n) = 1 \tag{10}$$

(6): 公钥 $KU = (e,d)$, 私钥 $KR = (d,n)$

公钥 KU:

$$\left\{\begin{array}{l} n: \text{素数} p \text{和} q \text{的乘积} \\ e: \text{与}(p-1)*(q-1) \text{互质} \end{array}\right\} \tag{11}$$

私钥 KR:

$$\left\{\begin{array}{l} n: \text{素数} p \text{和} q \text{的乘积} \\ d: \text{满足}(d*e) \bmod((p-1)*(q-1))=1 \end{array}\right\} \tag{12}$$

3.5.2 加解密流程

(1)加密过程:

$$C = M^e \bmod n \tag{13}$$

(2)解密过程:

$$M = C^d \bmod n \tag{14}$$

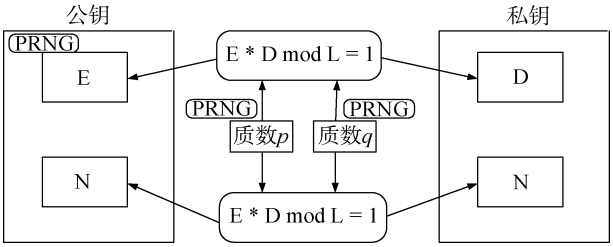

图 8 通行证加密方案

Figure 8 Token encryption scheme

4 方案安全性与性能分析

4.1 方案安全性分析

4.1.1 抗拒绝服务攻击

每个想要访问服务器数据的主体均需要向通行证属性管理中心 TAMC 申请通行证 token, 而不满足系统内应该具有的属性的主体则无法完成通行证 token 的申请。因此在数据访问阶段, 攻击者 Oscar 无法利用大量的僵尸主体访问数据库服务器, 能够保证每一位对数据库进行访问的主体都真实且可追踪溯源, 因此本方案可以抵抗拒绝服务攻击。

4.1.2 抗恶意篡改攻击

写入数据库服务器的每一条数据均会将数据地址 *info_address* 和数据哈希值 *info_hash* 写入数据链 Data_Chain 中, Data_Chain 中的主节点 Data_Chain_pri 将 *info_address* 和 *info_hash* 进行广播并部署到区块中。假设攻击者 Oscar 入侵数据库服务器并将数据进行修改, 首先日志链 Log_Chain 会将攻击者 Oscar 的入侵及篡改过程广播并记录到区块链上, 而后由于 Data_Chain 中的 *info_hash* 无法被篡改, 通过对比仍然可以判断出数据库服务器被入侵。

若攻击者 Oscar 想要攻击成功, 则必须控制超过 1/3 的节点, 由于在医疗系统中使用的区块链为联盟链, 加入节点具有身份认证机制, 因此攻击者理论上不可能控制超过 1/3 的节点进行攻击。所以本方案可以抗恶意篡改攻击。

4.1.3 抗单点失效攻击

跟传统的基于属性的访问控制方案相比, 本系统中日志部署点 LDP, 数据部署点 DDP, 策略部署点 PLP, 策略管理点 PAP,策略执行点 PEP, 策略决策点 PDP, 访问授权点 AAP 均以智能合约的形式部署在区块链上, 不需要人为调控, 只要达到特定条件便可以执行, 这样可以有效避免原方案中各点容易被收买或者攻击, 使访问授权过程发生错误。

另外, 访问授权日志 AAL 和数据哈希值 *info_hash* 分别存放在日志链 Log_Chain 和数据链 Data_Chain 中, 攻击者 Oscar 若只攻击一个节点或者一部分节点, 并不会使整个系统瘫痪。因此, 本方案可以抵抗失效攻击。

4.1.4 抗主体伪装攻击

攻击者 Oscar 向数据库服务器发送请求时需要提供自己的访问通行证 Access_U(Access_ID|Access Strategy|Access Attribute),若攻击者不通过正规渠道申请, 则无法获得访问通行证。

假设攻击者 Oscar 截到他人访问通行证, Access_U(Access_ID|AccessStrategy|Access Attribute)攻击者 Oscar 仍然无法进行数据访问。截取到的通行证属于加密状态, 其中内容均为加密数据 *Ciphertext_a* 该通行证需要利用通行证拥有者的私钥对通行证进行解密, 获得解密数据 *Plaintext_a*,才可以成功访问数据库服务器。若无法正确解密, 则无法使用截取到的通行证对数据进行访问。因此本方案可以抵抗主体伪装攻击。

4.1.5 抗重放攻击

主体S向通行证属性管理中心TAMC申请通行证时, 通行证属性管理中心 TAMC 会返回特定的 Token(Token_ID|Token_Data|Generation Time*)* 当攻击者 Oscar 截获主体 S 个人信息并进行重放时, 通行证属性管理中心 TAMC 判断主体信息并判断该身份的 *Generate Time* 和 *Expiration_Time*, 当 $C||T-G||T<E\,||\,T-G\,||\,T$ 时说明当前的系统存在重放攻击, 此时将拒绝申请。

若攻击者 Oscar 截取到访问通行证,并利用 Access_U(Access_ID|AccessStrategy|Access Attribute) 对数据库服务器 DB 进行访问时, 由于 *Access_ID* 是访问通行证的唯一控制标识符, 并且每一个访问通行证 Access_Token 只允许访问一次, 访问之后系统将记录 *Access_ID* 并进行保存, 因此当系统

判断出该 *Access_ID* 已经进行过访问时，便会拒绝攻击者 Oscar 的访问请求。综上，本系统可以抵抗重放攻击。

4.2 性能分析

4.2.1 实验分析

本文基于 Python 语言实现了属性通行证的 RSA 加解密，具体实验环境如下：操作系统为 Windows10 中文版 64 位，CPU 为 Inter(R)Core(TM)i7-8750H @2.20GHz，内存大小为 8GB 内存。

如图 9 所示，本文首先以访问次数为自变量，访问控制策略个数相同时所需访问时间为因变量，对访问次数分别为 5 次、10 次、15 次、20 次和 25 次时，本方案和 ABAC 方案以及方案[13]所需的访问时间进行了仿真和对比，表现出在相同访问控制策略的情况下访问次数越多，本方案的优势越明显。

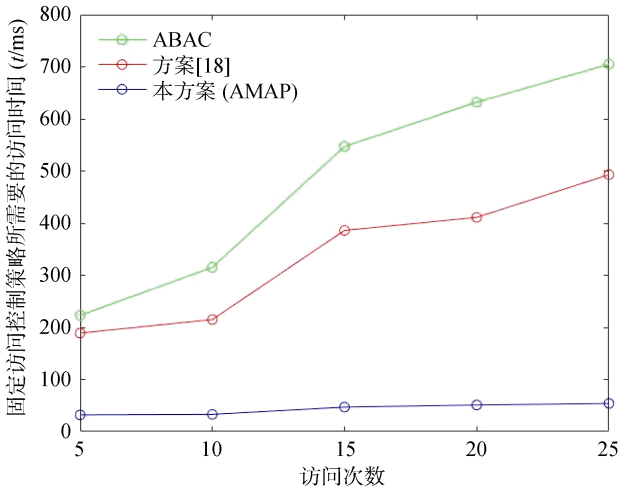

图 9 访问不同次数所需时间

Figure 9 Time required to visit different times

如图 10 所示，本文又以访问控制策略为自变量，访问次数相同时所需访问时间为因变量，对访问控制策略个数分别为 5 个、10 个、15 个、20 个和 25 个时，本方案与 ABAC 方案以及方案[13]所需的访问时间进行了仿真和对比,表现出在相同访问次数情况下访问控制策略个数越多时，本方案的优势越明显。

如表 3 所示，本文对 100s 中不同访问策略个数下，ABAC 方案和本方案访问次数吞吐量进行了对比，得到了在访问控制策略分别为 50、100、150、200 和 250 的情况下，本方案的吞吐量相比 ABAC 方案来讲均有较大的优势。

如图 11 所示，本方案分别对长度为 128 位，256 位，512 位，1024 位的密钥对生成时间进行了实验分析，其中横坐标为密钥长度，单位(bit)，纵坐标代表时间，单

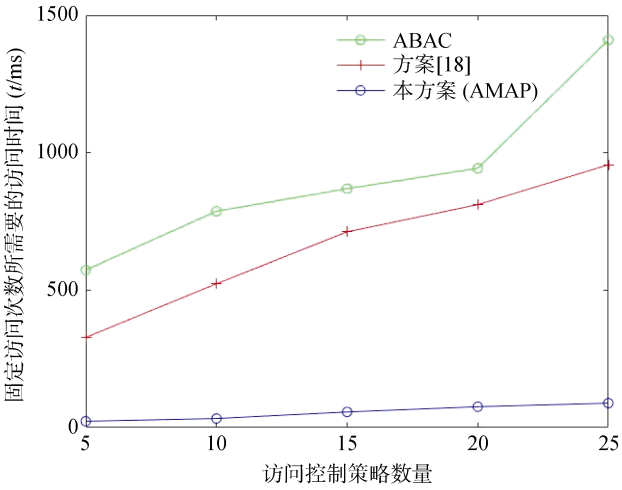

图 10 访问策略个数不同时的访问时间

Figure 10 Access time when the number of access policies is different

表 3 ABAC 方案与本方案对比

Table 3 Comparison of ABAC scheme and this scheme

访问策略数量	ABAC 方案在 100s 内的访问次数	AMAP 方案在 100s 内的访问次数
50	7082	113636
100	4496	80645
150	3546	46296
200	2042	32467
250	1583	35510

位(s)。得到所需时间最长的 1024 位密钥对的生成仅需 0.3s。密钥对的生成速度符合现在医疗环境下对医疗数据使用的高效性需求。

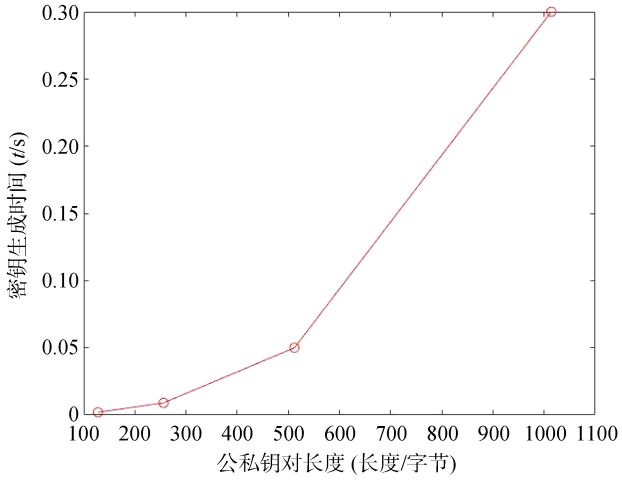

图 11 密钥生成所需时间

Figure 11 Time required for encryption key

在图 12，图 13 中，本方案分别以公钥长度和所需加密的明文长度为自变量，加密时间为因变量，判断在

本系统在不同公钥长度(分别是 128 位, 256 位, 512 位, 1024 位)和不同明文长度(分别是 128 位, 256 位, 512 位, 1024 位, 2048 位)时的加密速度, 在得到的 20 组数据中, 利用 1024 位公钥加密 2048 位明文时仅需 0.0011s。

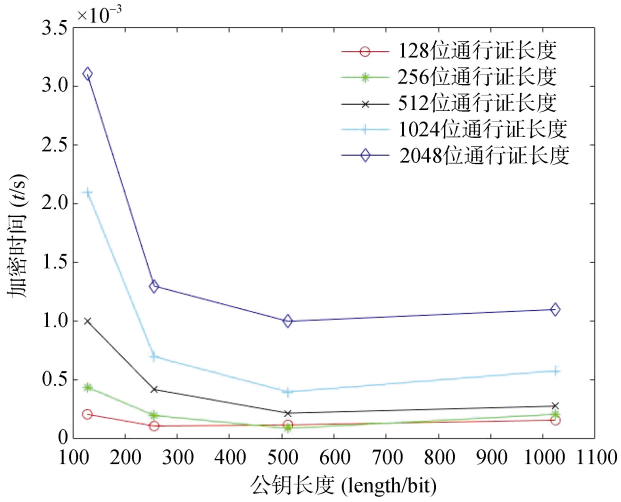

图 12 不同密钥及通行证长度下加密所需时间

Figure 12 Encryption time required under different keys and pass lengths

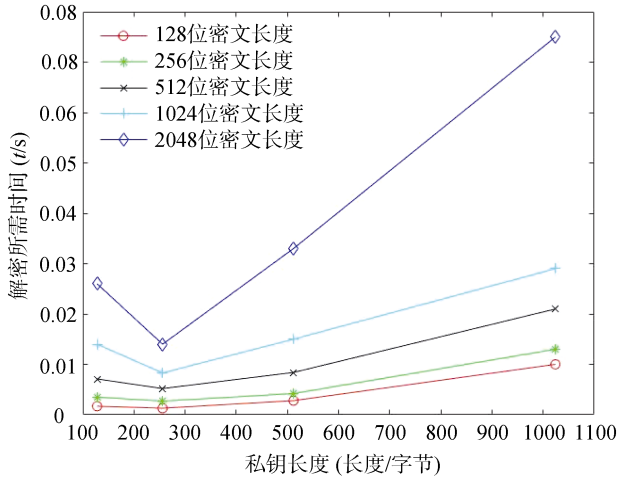

图 13 不同密钥及通行证长度下解密所需时间

Figure 13 Decryption time required under different keys and pass lengths

而在图 14, 图 15 中, 方法与上述类似, 在得到的 20 组解密密文的数据中, 利用 1024 位私钥解密 2048 位明文时仅需 0.075s。

上述的密钥对生成速度以及在尽可能保证数据安全性条件下的数据加解密速度当下医疗环境对医疗数据使用的高效性需求。

4.2.2 性能对比分析

表 4 将本方案与其他相关方案进行了对比, 从表 4 中可以发现, 方案[4]无法抵抗恶意篡改攻击和单点失效攻击, 方案[15]无法抵抗用户伪装攻击, 方案[18]无法抵抗用户伪装攻击, 方案[5]则对于表中提到的攻击均无法抵抗。而本文所提方案能够抵抗恶意篡改攻击, 拒绝服务攻击, 用户伪装攻击等, 与其他方案相比, 本方案对于相关攻击的抵抗有一定的优势。

表 5 将本方案与其他相关方案在是否可溯源, 访问是否公开透明等功能上进行了比较, 其中方案[4], [18]虽然实现了细粒度的访问控制, 但访问过程却无法公开且不可溯源。而方案[5], [15]也同样无法实现访问过程公开透明可溯源, 同时方案[5]也没有实现细粒度的访问控制。相比而言, 本文所提方案能够实现细粒度访问控制, 且访问公开透明可溯源, 在访问效率上也有一定优势。因此相比其他方案而言, 本方案在相关功能性上优势较明显。

表 4 本方案与其他方案抵抗攻击类型对比

Table 4 Comparison of attack types between this scheme and other schemes

抵抗攻击类型	[4]	[5]	[15]	[18]	AMAP
恶意篡改攻击	×	×	√	√	√
拒绝服务攻击	√	×	√	×	√
用户伪装攻击	√	×	×	√	√
重放攻击	√	×	√	√	√
单点失效攻击	×	×	√	√	√

表 5 本方案与其他方案功能性比较

Table 5 Functional comparison of this scheme and other schemes

功能	[4]	[5]	[15]	[18]	AMAP
可溯源	×	×	√	×	√
访问公开透明	√	×	√	√	√
细粒度访问控制	√	×	×	√	√
访问效率	√	×	√	√	√
基于 ABAC	×	×	√	√	√
自主授权	√	×	√	√	√

5 结束语

为了实现患者隐私数据的保护和必要数据的安全共享, 且保证访问控制过程公开可溯源, 本文以基于属性的访问控制方案为基础, 同时引入主体属性通行证, 提出了一种基于区块链和属性通行证的可溯源细粒度访问控制方案, 其中通过将区块链技术与基于属性的访问控制方案相结合, 通过将访问和授权日志上链, 在实现细粒度访问控制的同时, 支持访问控制过程的公开可溯源。同时, 将隐私数据摘要值以及数据地址上链, 实现了数据的不可篡改, 提高了数据的访问效率。另外, 本文提出的属性通

行证方案, 在保证访问安全的同时, 解决了传统模型中每次访问均需进行属性及策略匹配的问题, 提高了系统访问效率。最后, 本文通过安全性分析, 性能分析以及实验仿真, 表明了本方案的安全性和高效性。

参考文献

[1] Ju J H, Wu J Y, Fu J Q, et al. A Survey on Cloud Storage[J]. *Journal of Computers*, 2011, 6(8): 1764-1771.

[2] Fang L, Yin L H, Guo Y C, et al. A Survey of Key Technologies in Attribute-Based Access Control Scheme[J]. *Chinese Journal of Computers*, 2017, 40(7): 1680-1698.
(房梁, 殷丽华, 郭云川, 等. 基于属性的访问控制关键技术研究综述[J]. *计算机学报*, 2017, 40(7): 1680-1698.)

[3] Armbrust M, Fox A, Griffith R, et al. A View of Cloud Computing[J]. *Communications of the ACM*, 2010, 53(4): 50-58.

[4] Zhang Y T, Fu Y C, Yang M, et al. Access Control Scheme for Medical Data Based on PBAC and IBE[J]. *Journal on Communications*, 2015, 36(12): 200-211.
(张怡婷, 傅煜川, 杨明, 等. 基于 PBAC 模型和 IBE 的医疗数据访问控制方案[J]. *通信学报*, 2015, 36(12): 200-211.)

[5] Deng X J, Shen J N, Xu Z W, et al. Design and Implementation of Medical Cloud Platform Based on RBAC Model[J]. *Journal of Hubei Minzu University* (*Natural Science Edition*), 2020, 38(1): 93-97.
(邓学剑, 沈济南, 许振武, 等. 基于 RBAC 模型的医疗云平台的设计与实现[J]. *湖北民族大学学报(自然科学版)*, 2020, 38(1): 93-97.)

[6] Liu M, Luo Y, Yang C, et al. Privacy-Preserving Matrix Product Based Static Mutual Exclusive Roles Constraints Violation Detection in Interoperable Role-Based Access Control[J]. *Future Generation Computer Systems*, 2020, 109: 457-468.

[7] Figueroa-Lorenzo S, Añorga J, Arrizabalaga S. A Role-Based Access Control Model in Modbus SCADA Systems. a Centralized Model Approach[J]. *Sensors* (*Basel*, *Switzerland*), 2019, 19(20): 4455.

[8] Miao T T, Yang H J, Shen J. An Access Control Scheme with User Privacy Protection in E-Health Environment[J]. *Cyberspace Security*, 2019, 10(10): 16-22.
(苗田田, 杨惠杰, 沈剑. 电子医疗环境中支持用户隐私保护的访问控制方案[J]. *网络空间安全*, 2019, 10(10): 16-22.)

[9] Li J J, Yuan Y, Wang F Y. Blockchain-Based Digital Currency: The State of the Art and Future Trends[J]. *Acta Automatica Sinica*, 2021, 47(4): 715-729.
(李娟娟, 袁勇, 王飞跃. 基于区块链的数字货币发展现状与展望[J]. *自动化学报*, 2021, 47(4): 715-729.)

[10] Nakamoto S, Bitcoin A. A peer-to-peer electronic cash system [EB/OL]. Bitcoin. URL: https: //bitcoin.org/bitcoin.pdf, 2008.

[11] Li H, Zhang M, Feng D G, et al. Research on Access Control of Big Data[J]. *Chinese Journal of Computers*, 2017, 40(1): 72-91.
(李昊, 张敏, 冯登国, 等. 大数据访问控制研究[J]. *计算机学报*, 2017, 40(1): 72-91.)

[12] di Francesco Maesa D, Mori P, Ricci L. Blockchain Based Access Control[M]. Distributed Applications and Interoperable Systems. Cham: Springer International Publishing, 2017: 206-220.

[13] Thwin T T, Vasupongayya S. Blockchain-Based Access Control Model to Preserve Privacy for Personal Health Record Systems[J]. *Security and Communication Networks*, 2019, 2019: 1-15.

[14] Yang X D, Li T, Pei X Z, et al. Medical Data Sharing Scheme Based on Attribute Cryptosystem and Blockchain Technology[J]. *IEEE Access*, 8: 45468-45476.

[15] Ge J H, Shen T. Energy Data Access Control Method Based on Blockchain[J]. *Journal of Computer Applications*, 2021, 41(9): 2615-2622.
(葛纪红, 沈韬. 基于区块链的能源数据访问控制方法[J]. *计算机应用*, 2021, 41(9): 2615-2622.)

[16] Zyskind G, Nathan O, Pentland A ', et al. Decentralizing privacy: using blockchain to protect personal data[C]. *2015 IEEE Security and Privacy Workshops*, 2015: 180-184.

[17] Neisse R, Steri G, Nai-Fovino I. A Blockchain-Based Approach for Data Accountability and Provenance Tracking[C]. *The 12th International Conference on Availability*, *Reliability and Security*, 2017: 1-10.

[18] Liu A D, Du X H, Wang N, et al. Blockchain-Based Access Control Mechanism for Big Data[J]. *Journal of Software*, 2019, 30(9): 2636-2654.
(刘敖迪, 杜学绘, 王娜, 等. 基于区块链的大数据访问控制机制[J]. *软件学报*, 2019, 30(9): 2636-2654.)

[19] Bhajantri L B, Mujawar T N. A Comprehensive Review of Access Control Mechanism Based on Attribute Based Encryption Scheme for Cloud Computing[J]. *International Journal of Advanced Pervasive and Ubiquitous Computing*, 2019, 11(3): 33-52.

[20] Hemdi M, Deters R, Communication N A B T, et al. Using REST based protocol to enable ABAC within IoT systems[C]. *2016 IEEE 7th Annual Information Technology*, *Electronics and Mobile Communication Conference*, 2016: 1-7.

[21] Xie R N, Li H, Shi G Z, et al. Blockchain-Based Access Control Mechanism for Data Traceability[J]. *Journal on Communications*, 2020, 41(12): 82-93.
(谢绒娜, 李晖, 史国振, 等. 基于区块链的可溯源访问控制机制[J]. *通信学报*, 2020, 41(12): 82-93.)

[22] Wang H Q, Wu T. Cryptography on the Blockchain[J]. *Journal of Nanjing University of Posts and Telecommunications* (*Natural Science Edition*), 2017, 37(6): 61-67.
(王化群, 吴涛. 区块链中的密码学技术[J]. *南京邮电大学学报(自然科学版)*, 2017, 37(6): 61-67.)

[23] Zhang C, Li Q, Chen Z H, et al. Medical Chain: Alliance Medical Blockchain System[J]. *Acta Automatica Sinica*, 2019, 45(8): 1495-1510.
(张超, 李强, 陈子豪, 等. Medical Chain: 联盟式医疗区块链系统[J]. *自动化学报*, 2019, 45(8): 1495-1510.)

[24] Harrison C, Eckman B, Hamilton R, et al. Foundations for Smarter Cities[J]. *IBM Journal of Research and Development*, 2010, 54(4): 1-16.

[25] Nam T, Pardo T A. Conceptualizing Smart City with Dimensions of Technology, People, and Institutions[C]. *The 12th Annual International Digital Government Research Conference*: *Digital Government Innovation in Challenging Times*, 2011:

282-291.

[26] Kong Y, Zhao J S, Yuan L, et al. Research on data sharing analysis and key technology of smart city[C]. *2018 26th International Conference on Geoinformatics*, 2018: 1-7.

罗维 于 2019 年在西安电子科技大学密码学专业获得博士学位。现任西安邮电大学网络空间安全学院讲师, 研究领域为公钥密码学、云存储安全。研究兴趣包括: 智慧医疗、车联网。Email: rovid008@163.com

韩刚 于 2020 年在西北工业大学电子科学与技术专业获得博士学位。现任西安邮电大学网络空间安全学院讲师, 研究领域为公钥密码学、属性基加密、区块链技术。研究兴趣包括: 智慧医疗、智慧农业。Email: hangang668866@163.com

吕英泽 现在西安邮电大学网络空间安全学院信息安全专业攻读学士学位, 研究领域为区块链技术、属性基加密。研究兴趣包括: 智慧医疗。Email: lvyingze0809@163.com

王嘉乾 现在西安邮电大学网络空间安全学院信息安全专业攻读学士学位, 研究领域为公钥密码学、云计算、区块链技术。研究兴趣包括: 智慧医疗。Email: wangjq990817@126.com